

Insect survey 2020, basic analysis

Jens Å

29 October, 2020

Contents

Data import and cleaning	2
Filter out the Nerlandsøya samples	4
Check the emptying dates.	4
Check missing weight data	7
Basic species richness graphs	8
Basic biomass graphs	10
Check overlap between malaise and window traps	14
Total number of coleoptera species per trap type	14
Percentage of catches in each trap type per species	15
Figs to Niclas on Pollinators	19
Phenology graphs	21
Plot some weather data	25
Check correspondance with altitude	29
To-do	
• These species are not matched to the gyldig names in artsdatabasen yet.	
• Should store dates and times separately as well in logger_data table. Easier to subset later.	
• Update weight figures with bottle weight removed.	
• Make plots of phenology	
– No. species over the season for each location. Split into separate figs for the two habitats?	
– Relate this to local temperature, temperature sums?	
• Plots of number of samples over time	
• Check differences btw 2 week and 4 week sampling	
– Species occurrences,	

- compare 2 * 2 weeks vs. 1 * 4 week
- Compare first and second of the 2 weeks, see if any of the have more overlap with 4 week. If 4 week becomes full, the first 2 week period should overlap more.
- Do this for window traps as well.
- Check overlap/additions of window traps vs. malaise traps
 - Subset only Coleoptera
 - Group by location + time
- Within habitat, check relationship btw species richness and a few variables
 - Forest age
 - Diversity of fauna
 - Temperature
 - Altitude
- Check differences btw ethanol and propglyk window traps

Data import and cleaning

At this point I get the raw data from Marie, which requires some cleaning. In the future, we'll put this all in a database, and separate out the cleaning and database import in a separate script.

```
#dat <- read_delim("../Data/Genetikkdata/long_format_data_GCF-2020-735_736_19102020.txt",
#  as.tibble()

#dat <- read_delim("../Data/Genetikkdata/long_format_data_GCF-2020-735_736_19102020.txt"

dat <- read_delim("../Data/Genetikkdata/long_format_data_GCF-2020-735_736_744_26102020.txt",
  delim = ";",
  col_types = cols(
    kingdom = col_character(),
    phylum = col_character(),
    class = col_character(),
    order = col_character(),
    family = col_character(),
    genus = col_character(),
    species = col_character(),
    sample = col_character(),
    value = col_double(),
    GenlabID = col_character(),
    GUID = col_character(),
    Tilsatt_antall_C.analis_SI_USA_08.06.20 = col_double(),
    Tilsatt_antall_C.chinensis_08.06.20 = col_double(),
```

```

Tilsatt_antall_Sirisser = col_double(),
Tilsatt_antall_Melormer = col_double(),
Tilsatt_antall_C.maculatus = col_double(),
Vekt_tom_flaske_.g. = col_double(),
Vekt_flaske_med_EtOH_og_insekter_.g. = col_double(),
Vekt_flaske_.insekter_vaat_.g. = col_double(),
Vekt_flaske_.insekter_toerr_.g. = col_double(),
Totalvolum_.mengde_ATL_.prot.K_mL. = col_character(),
Vaatvekt_.g. = col_double(),
Toerrvekt_.g. = col_double(),
Kommentar_proeve = col_character(),
Ekstraksjonsdato = col_date(format = "%d.%m.%Y"),
Ekstraksjonskit = col_character(),
Subsamplet_volum_.ul. = col_double(),
Elueringsvolum = col_double(),
OD = col_double(),
X260_280 = col_double(),
X260_230 = col_double(),
Labperson = col_character(),
Ekstraksjonskommentar = col_character(),
locality = col_character(),
trap = col_character(),
date_placed = col_date(format = "%d.%m.%Y"),
time_placed = col_time(format = ""),
empty_week = col_double(),
dateemptied = col_date(format = "%d.%m.%Y"),
timeemptied = col_time(format = ""),
liquid = col_character(),
personemptied = col_character(),
sample_id = col_character(),
mottat_lab = col_character(),
comment = col_character()
),
na = c("NA",
      "N/A",
      "Maa veies"
)
)

```

Fix some coding errors.

```

# dat %>%
#   select(genus, species, new_species) %>%
#   slice(297, 1902)
#
#

```

```

# dat %>%
#   select(genus, species, new_species) %>%
#   slice(3219, 3585, 4017, 11424, 13275, 15465, 21056, 24168, 25077, 31567, 32805, 48502, 49
#
#Ekstraksjonskommentar = gsub(pattern = "[\\\"]", "", Ekstraksjonskommentar),
#
dat <- dat %>%
  mutate(species_latin = species)
#
dat <- dat %>%
  mutate(species = gsub(" sp", "_sp", species))
#
dat <- dat %>%
  mutate(species = gsub("[0-9](_)(sp)", "\\\\[1]\\\[3", species))
#
dat <- dat %>%
  mutate(species = gsub("(.*)(_)(.*)", "\\\\[3", species))

```

Set factor levels

```

dat <- dat %>%
  mutate(locality = factor(locality, levels = c(paste0("semi-nat_", 1:10), paste0("skog_", 1
trap_type = ifelse(grepl("MF", trap), "Malaise", "Vindu"),
habitat_type = ifelse(grepl("skog", locality), "Skog", "SN_Gressmark"),
liquid = ifelse(grepl("eto", liquid), "Ethanol", "PropGl_Ethanol"),
weeks_sampled = ifelse(grepl("1", trap) | grepl("3", trap), 2, 4)
)

```

Filter out the Nerlandsøya samples

```

dat <- dat %>%
  filter(locality != "Nerlandsoeya") %>%
  droplevels()

```

Check the emptying dates.

The dates seems to be OK this time. Should do a proper check on the endpoints though.

```

dat <- dat %>%
  mutate(days_collected = as.numeric(dat$date_emptied - dat$date_placed))

# dat %>%

```

```

#   filter(days_collected == 0) %>%
#   select(sample_id,
#         GUID,
#         date_placed,
#         dateemptied) %>%
#   distinct() %>%
#   print(n = Inf)
#
# hist(as.numeric(dat$dateemptied - dat$date_placed))

dat %>%
  group_by(days_collected) %>%
  summarise(n())

## `summarise()` ungrouping output (override with `.`groups` argument)

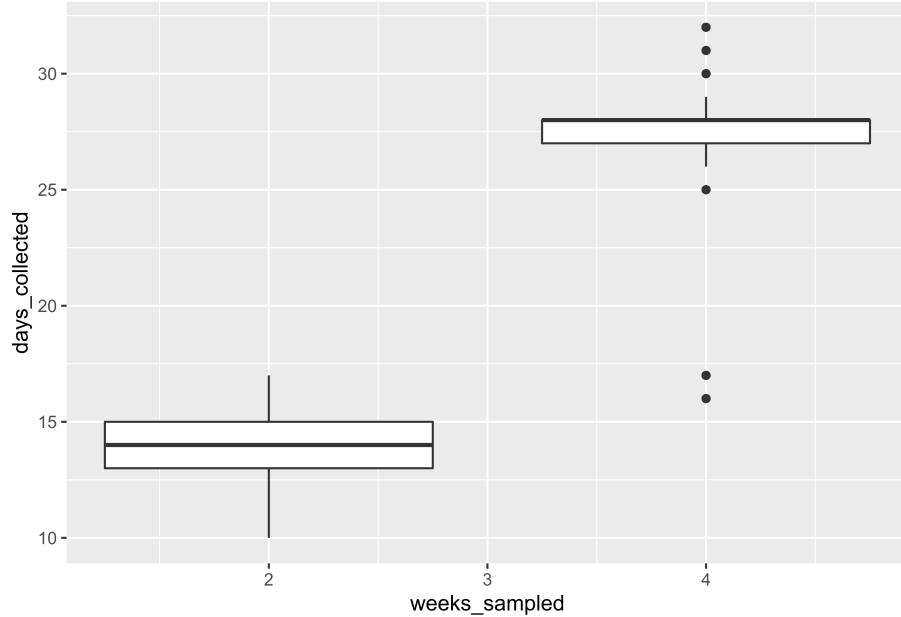
## # A tibble: 16 x 2
##   days_collected `n()`
##   <dbl> <int>
## 1 10     368
## 2 11    1084
## 3 12    5654
## 4 13    6223
## 5 14   10831
## 6 15    8053
## 7 16   11159
## 8 17    1422
## 9 25    561
## 10 26   1101
## 11 27   3016
## 12 28   9687
## 13 29   1132
## 14 30    450
## 15 31   1175
## 16 32    734

# dat %>%
#   filter(days_collected == 8) %>%
#   select(sample_id,
#         GUID,
#         date_placed,
#         dateemptied) %>%
#   distinct() %>%
#   print(n = Inf)

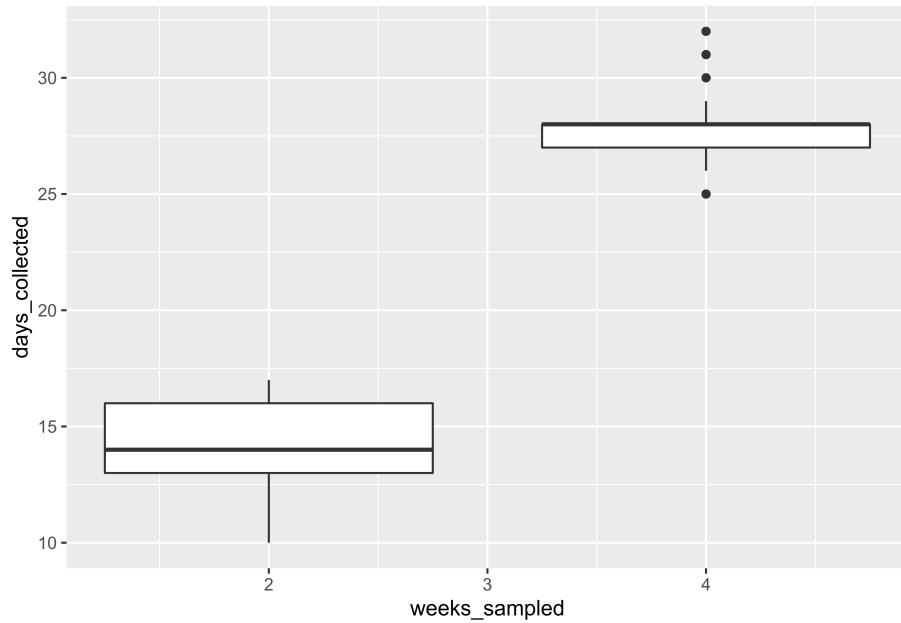
```

Set weeks sampled manually for the few locations in Oslo that got an extra short sampling in the beginning.

```
ggplot(dat, aes(x = weeks_sampled, y = days_collected, group = weeks_sampled)) +  
  geom_boxplot()
```



```
twoWeeks <- dat %>%  
  filter(weeks_sampled == 4 &  
         days_collected < 20) %>%  
  select(sample_id) %>%  
  distinct() %>%  
  as.vector()  
  
dat <- dat %>%  
  mutate(weeks_sampled = ifelse(sample_id %in% twoWeeks[[1]], 2, weeks_sampled))  
  
ggplot(dat, aes(x = weeks_sampled, y = days_collected, group = weeks_sampled)) +  
  geom_boxplot()
```



Check missing weight data

```
dat %>%
  group_by(empty_week) %>%
  select(Vekt_tom_flaske_.g.,
         Vekt_flaske_med_EtOH_og_insekter_.g.,
         Vekt_flaske_.insekter_vaat_.g.,
         Vekt_flaske_.insekter_toerr_.g.,
         Vaatvekt_.g.,
         Toerrvekt_.g.
  ) %>%
  distinct() %>%
  summarise_all(~sum(is.na(.)))

## Adding missing grouping variables: `empty_week`
## # A tibble: 6 x 7
##   empty_week Vekt_tom_flaske~ Vekt_flaske_med~ Vekt_flaske_.i~ Vekt_flaske_.i~
##   <dbl>          <int>          <int>          <int>          <int>
## 1 22            6              0              0              0
## 2 24            20             0              0              0
## 3 26            19             3              0              2
## 4 28            4              9              0              0
```

```

## 5          30          2          2          0          2
## 6          32          0          0          0          0
## # ... with 2 more variables: Vaatvekt_.g. <int>, Toerrvekt_.g. <int>

```

Basic species richness graphs

Quick look at diversity at the different sites.

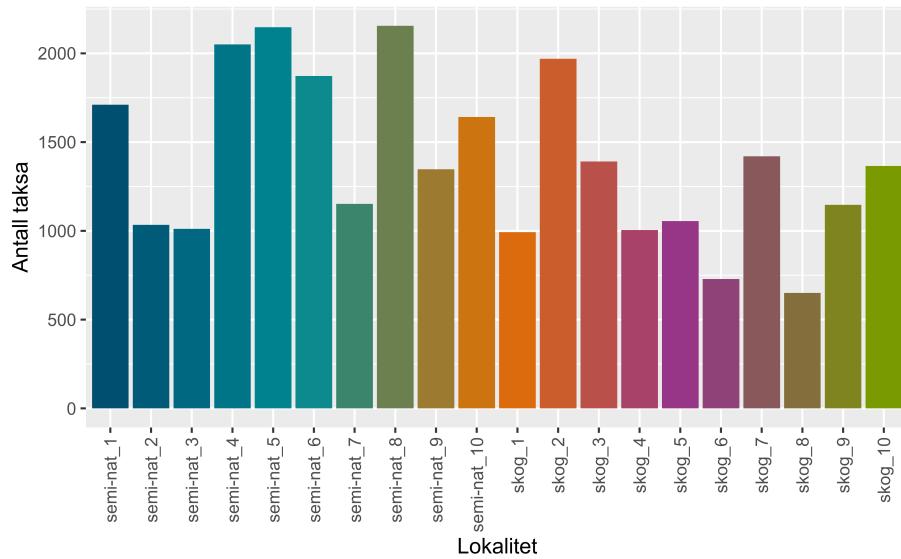
```

total_no_taxa <- dat %>% select(genus,species) %>%
  distinct() %>%
  count()

dat %>%
  filter(grepl("MF", trap)) %>%
  group_by(locality) %>%
  select(genus,species) %>%
  distinct() %>%
  count() %>%
  ungroup() %>%
  ggplot(.) +
  geom_bar(aes(y = n, x = locality, fill = locality),
            stat = "identity") +
  ggtitle(paste0("Kumulativ artsrikedom i\nalle malaisefeller per ", max(dat$date_emptied)))
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1),
        legend.position = "none") +
  scale_fill_nina() +
  ylab("Antall taksa") +
  xlab("Lokalitet")

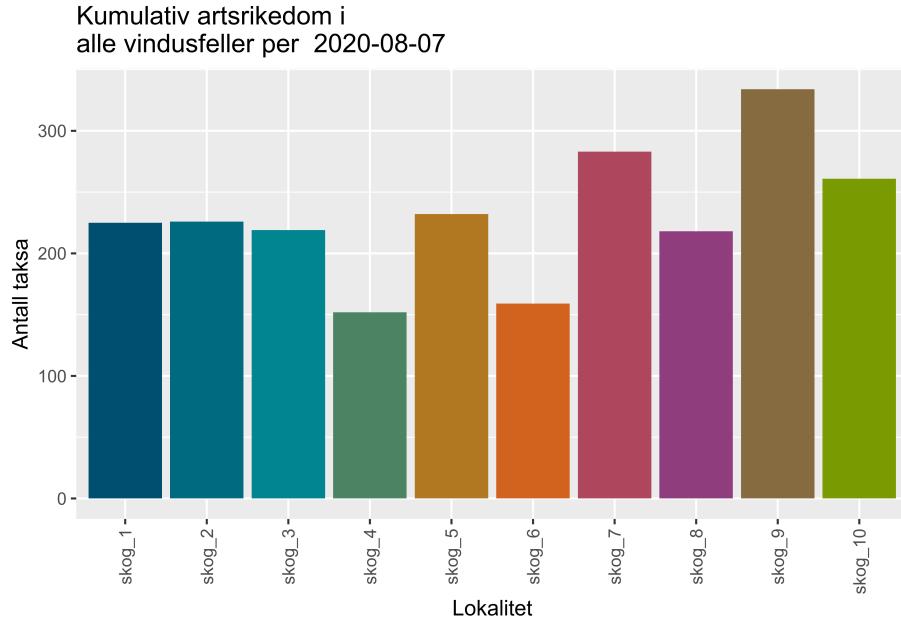
## Adding missing grouping variables: `locality`
```

Kumulativ artsriksdom i
alle malaisefeller per 2020-08-07



```
dat %>%
  filter(grepl("VF", trap)) %>%
  group_by(locality) %>%
  select(genus, species) %>%
  distinct() %>%
  count() %>%
  ungroup() %>%
  ggplot(.) +
  geom_bar(aes(y = n, x = locality, fill = locality),
           stat = "identity") +
  ggtitle(paste0("Kumulativ artsriksdom i\\nalle vindusfeller per ", max(dat$dateemptied)))
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1),
        legend.position = "none") +
  scale_fill_nina() +
  ylab("Antall taksa") +
  xlab("Lokalitet")

## Adding missing grouping variables: `locality`
```



Basic biomass graphs

NB, we still have some missing data on the weight of the bottles.

Quick quality check of the spread of the bottle weights. Could we impute the missing values?

```
dat %>%
  select(Vekt_tom_flaske_.g.,
         Vekt_flaske_med_EtOH_og_insekter_.g.,
         Vekt_flaske_._insekter_vaat_.g.,
         Vekt_flaske_._insekter_toerr_.g.,
         Vaatvekt_.g.,
         Toerrvekt_.g.)

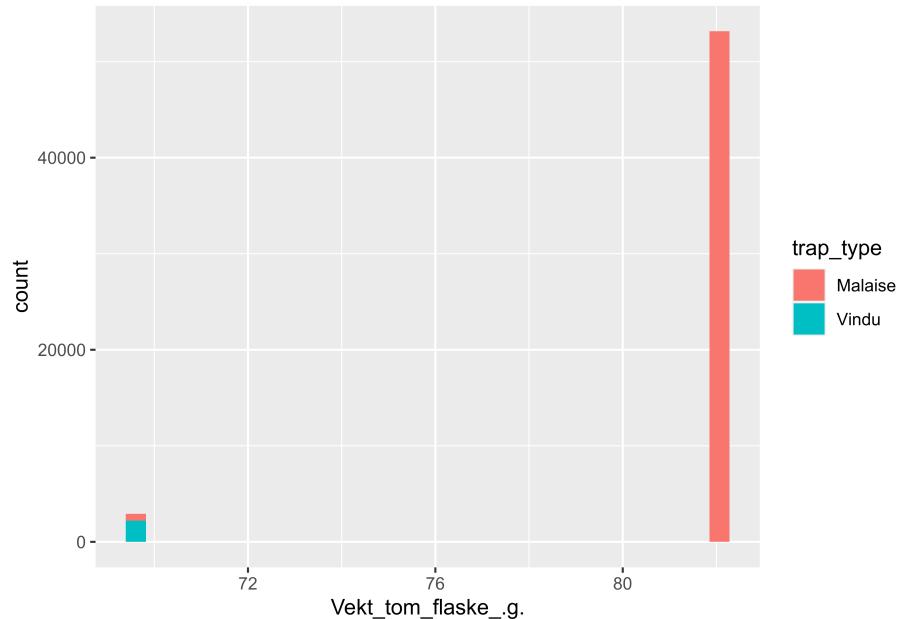
## # A tibble: 62,650 x 6
##   Vekt_tom_flaske~ Vekt_flaske_med~ Vekt_flaske_..i~ Vekt_flaske_..i~
##   <dbl>           <dbl>           <dbl>           <dbl>
## 1 82.2            728             118.            89.0
## 2 82.2            728             118.            89.0
## 3 82.2            728             118.            89.0
## 4 82.2            728             118.            89.0
## 5 82.2            728             118.            89.0
## 6 82.2            728             118.            89.0
```

```

## 7           82.2          728          118.         89.0
## 8           82.2          728          118.         89.0
## 9           82.2          728          118.         89.0
## 10          82.2          728          118.         89.0
## # ... with 62,640 more rows, and 2 more variables: Vaatvekt_.g. <dbl>,
## #   Toerrvekt_.g. <dbl>
ggplot(dat) +
  geom_histogram(aes(Vekt_tom_flaske_.g., fill = trap_type))

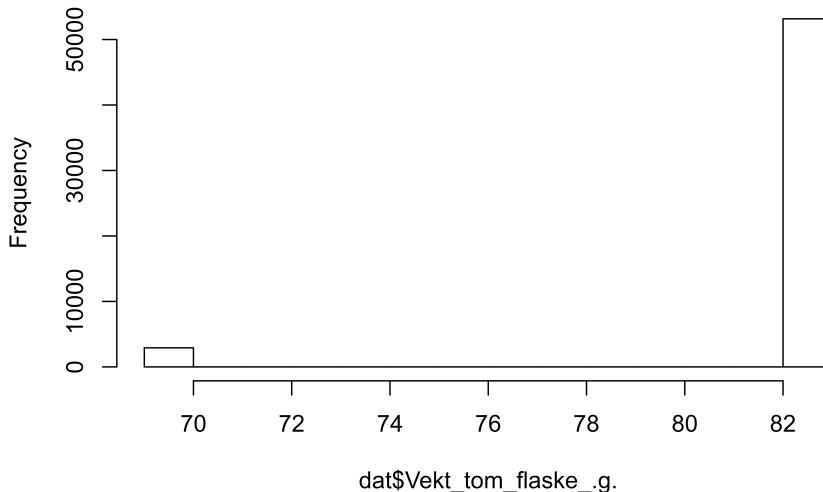
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## Warning: Removed 6572 rows containing non-finite values (stat_bin).

```



```
hist(dat$Vekt_tom_flaske_.g.)
```

Histogram of dat\$Vekt_tom_flaske_g.



```
table(dat$Vekt_tom_flaske_.g.)
```

```
##  
##   69.75   82.21 82.2103  
##   2915   40333   12830
```

These bottle weights are weirdly consistent. Did they really weigh all bottles?
UPDATE WITH NEW BOTTLE WEIGHTS.

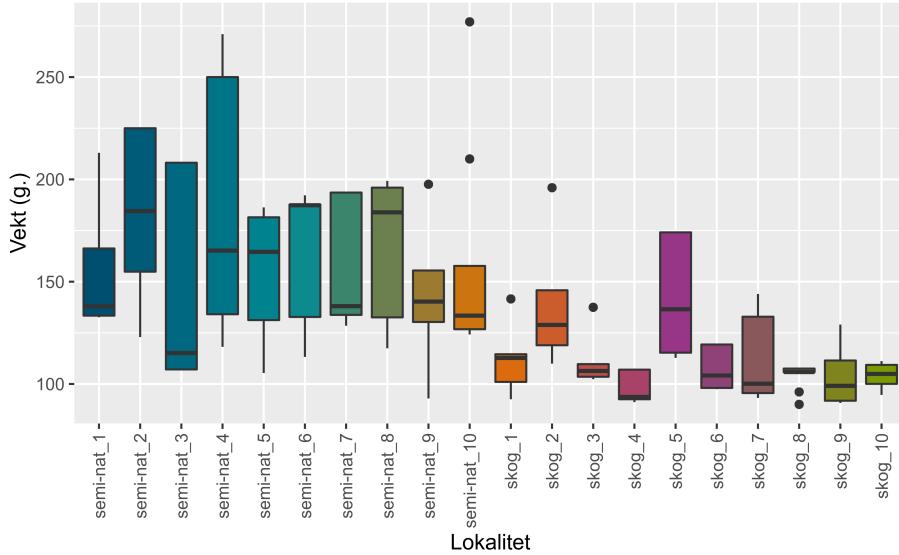
```
summary(dat$Toerrvekt_.g.)
```

```
##      Min. 1st Qu. Median     Mean 3rd Qu.    Max.    NA's  
## -0.570   4.740   8.328  13.463  16.170  80.350  9411
```

Quick look at the biomass at the different sites.

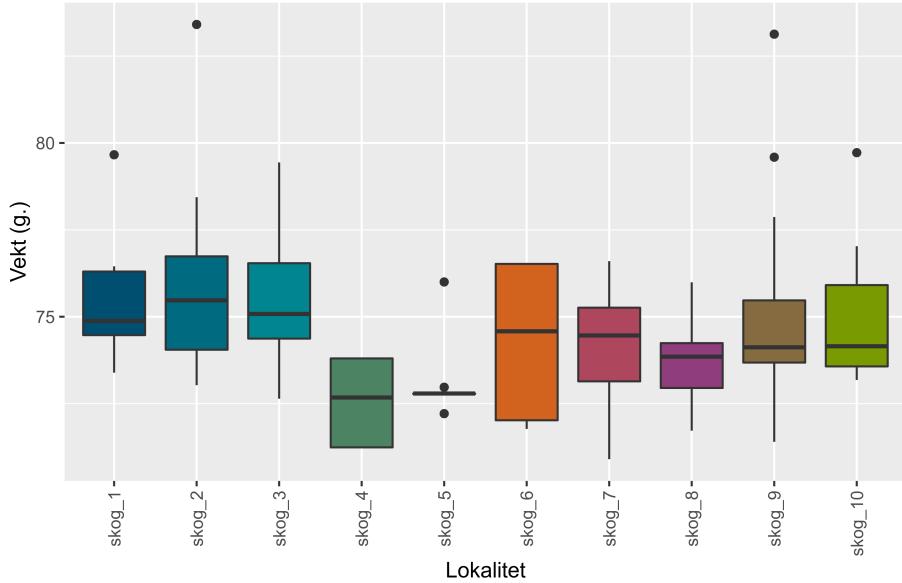
```
dat %>%  
  filter(grepl("MF", trap)) %>%  
  group_by(locality) %>%  
  ggplot(.) +  
  geom_boxplot(aes(y = Vekt_flaske_.insekter_vaat_.g., x = locality, fill = locality)) +  
  ggtile(paste0("Middels avrunnen vekt (pluss flaske)\nav insekter i malaisefeller per ",  
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1),  
  legend.position = "none") +  
  scale_fill_nina() +  
  ylab("Vekt (g.)") +  
  xlab("Lokalitet")
```

Middels avrunnen vekt (pluss flaske)
av insekter i malaisefeller per 2020-08-07



```
dat %>%
  filter(grepl("VF", trap)) %>%
  group_by(locality) %>%
  ggplot(.) +
  geom_boxplot(aes(y = Vekt_flaske_.insekter_vaat_.g., x = locality, fill = locality)) +
  ggttitle(paste0("Middels avrunnen vekt (pluss flaske) av insekter i vindusfeller per ", max))
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1),
        legend.position = "none") +
  scale_fill_nina() +
  ylab("Vekt (g.)") +
  xlab("Lokalitet")
```

Middels avrunnen vekt (pluss flaske) av insekter i vindusfeller per 2020-08.



Check overlap between malaise and window traps

Simple: total spec richn for all traps and times per location, plus the same for just malaise and window traps. group in figure by location

complex: tally how many spec is found in only one trap type. No unnecessary per species: which sp is only found in one trap type. or per species, % of findings in each trap type

Total number of coleoptera species per trap type

```
col_spec_richn_tot <- dat %>%
  filter(order == "Coleoptera",
         habitat_type == "Skog") %>%
  group_by(locality) %>%
  summarize(col_spec_richn = n_distinct(genus, species)) %>%
  mutate(trap_type = "Alle")

## `summarise()` ungrouping output (override with `groups` argument)
col_spec_richn_trap <- dat %>%
  filter(order == "Coleoptera",
         habitat_type == "Skog") %>%
  group_by(locality) %>%
  summarise(species = n_distinct(species)) %>%
  mutate(trap_type = "Alle")
```

```

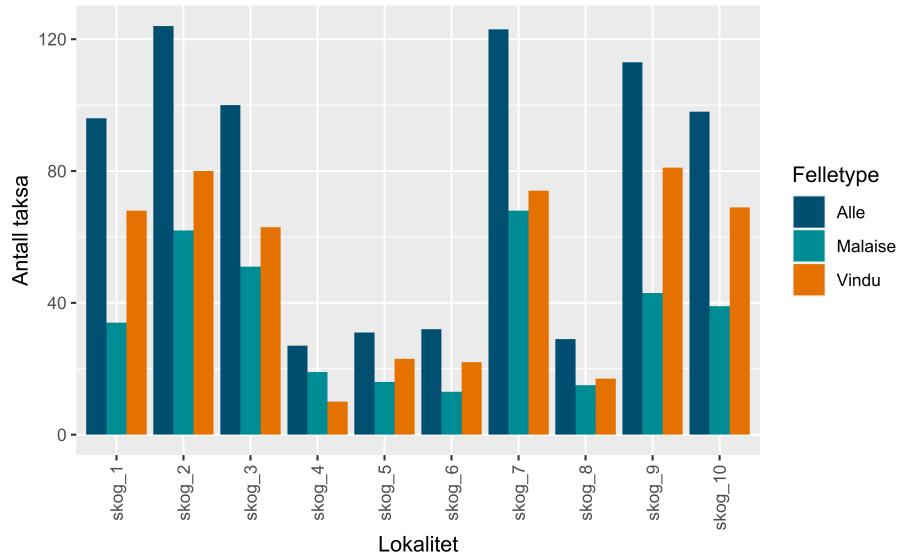
    habitat_type == "Skog") %>%
group_by(locality,
         trap_type) %>%
summarize(col_spec_richn = n_distinct(genus, species))

## `summarise()` regrouping output by 'locality' (override with `groups` argument)
col_spec_richn <- col_spec_richn_tot %>%
  union_all(col_spec_richn_trap)

ggplot(col_spec_richn, aes(y = col_spec_richn,
                           x = locality,
                           fill = trap_type)) +
  geom_bar(stat = "identity",
            position = position_dodge()) +
  ggtitle(paste0("Kumulativt antall billearter (identifiserte taksa)\ni hver felletyp per
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1)) +
  scale_fill_nina(name = "Felletype") +
  ylab("Antall taksa") +
  xlab("Lokalitet")

```

Kumulativt antall billearter (identifiserte taksa)
i hver felletyp per 2020-08-07



Percentage of catches in each trap type per species

How to make it more readable? Split out all the singletons into a separate graph?

```

col_perc_traptype <- dat %>%
  filter(order == "Coleoptera",
         habitat_type == "Skog") %>%
  group_by(trap_type,
           genus,
           species_latin) %>%
  summarize(no_catches = n())

## `summarise()` regrouping output by 'trap_type', 'genus' (override with `.`groups` argument)
p1 <- col_perc_traptype %>%
  ungroup() %>%
  arrange(species_latin,
          trap_type) %>%
  filter(no_catches > 1) %>%
  slice(1:50) %>%
  group_by(trap_type) %>%
  ggplot(.) +
  geom_bar(aes(y = reorder(species_latin, desc(species_latin)), x = no_catches, fill = trap_type),
            stat = "identity",
            position = position_dodge()) +
  scale_fill_nina(name = "Trap type") +
  ylab("") +
  xlab("Number of catches") +
  theme(legend.position = "none",
        text = element_text(size=5))

p2 <- col_perc_traptype %>%
  ungroup() %>%
  arrange(species_latin,
          trap_type) %>%
  filter(no_catches > 1) %>%
  slice(51:100) %>%
  group_by(trap_type) %>%
  ggplot(.) +
  geom_bar(aes(y = reorder(species_latin, desc(species_latin)), x = no_catches, fill = trap_type),
            stat = "identity",
            position = position_dodge()) +
  scale_fill_nina(name = "Trap type") +
  ylab("") +
  xlab("Number of catches") +
  theme(legend.position = "none",
        text = element_text(size=5))

p3 <- col_perc_traptype %>%

```

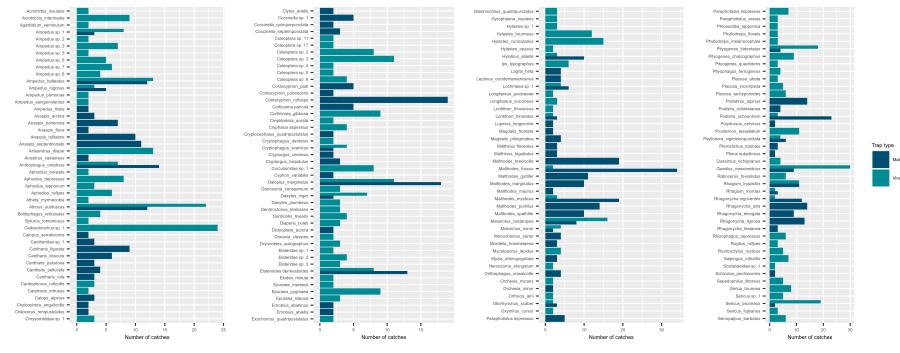
```

ungroup() %>%
arrange(species_latin,
        trap_type) %>%
filter(no_catches > 1) %>%
slice(101:150) %>%
group_by(trap_type) %>%
ggplot(.) +
  geom_bar(aes(y = reorder(species_latin, desc(species_latin)), x = no_catches, fill = trap_
                stat = "identity",
                position = position_dodge()) +
  scale_fill_nina(name = "Trap type") +
  ylab("") +
  xlab("Number of catches") +
  theme(legend.position = "none",
        text = element_text(size=5))

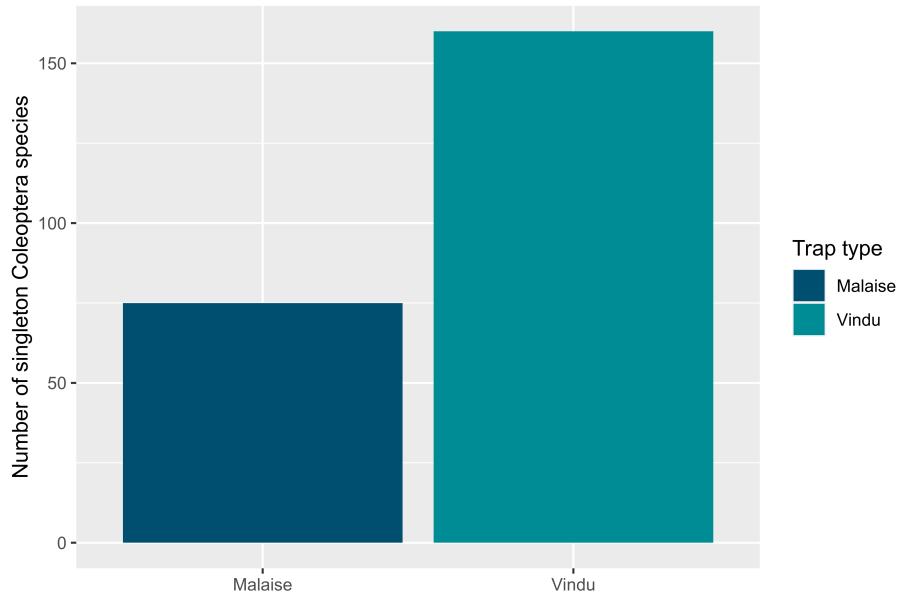
p4 <- col_perc_traptyle %>%
ungroup() %>%
arrange(species_latin,
        trap_type) %>%
filter(no_catches > 1) %>%
slice(151:200) %>%
group_by(trap_type) %>%
ggplot(.) +
  geom_bar(aes(y = reorder(species_latin, desc(species_latin)), x = no_catches, fill = trap_
                stat = "identity",
                position = position_dodge()) +
  scale_fill_nina(name = "Trap type") +
  ylab("") +
  xlab("Number of catches") +
  theme(text = element_text(size=5))

grid.arrange(p1, p2, p3, p4, nrow = 1)

```



```
col_perc_traptype %>%
  ungroup() %>%
  arrange(species_latin,
         trap_type) %>%
  filter(no_catches == 1) %>%
  group_by(trap_type) %>%
  summarize(no_singletons = n()) %>%
  ggplot(.) +
  geom_bar(aes(y = no_singletons, x = trap_type, fill = trap_type),
           stat = "identity") +
  scale_fill_nina(name = "Trap type") +
  ylab("Number of singleton Coleoptera species") +
  xlab("")
```



Figs to Niclas on Pollinators

Niclas Gyllenstrand asked how efficient the malaisetraps+metabarcoding are at finding pollinators. This could be a relevant graph for the report as well.

```
poll_spec_richn <- dat %>%
  filter(trap_type == "Malaise") %>%
  filter(family == "Andrenidae" |
         family == "Apidae" |
         family == "Colletidae" |
         family == "Halictidae" |
         family == "Megachilidae" |
         family == "Melittidae" |
         family == "Syrphidae" |
         family == "Hesperiidae" |
         family == "Lycaenidae" |
         family == "Nymphalidae" |
         family == "Papilionidae" |
         family == "Pieridae" |
         family == "Riodinidae")
  ) %>%
  group_by(family) %>%
```

```

distinct(family, genus, species) %>%
summarize(col_spec_richn = n(),
          no_unidentified_sp = sum(grepl("sp.", species)))

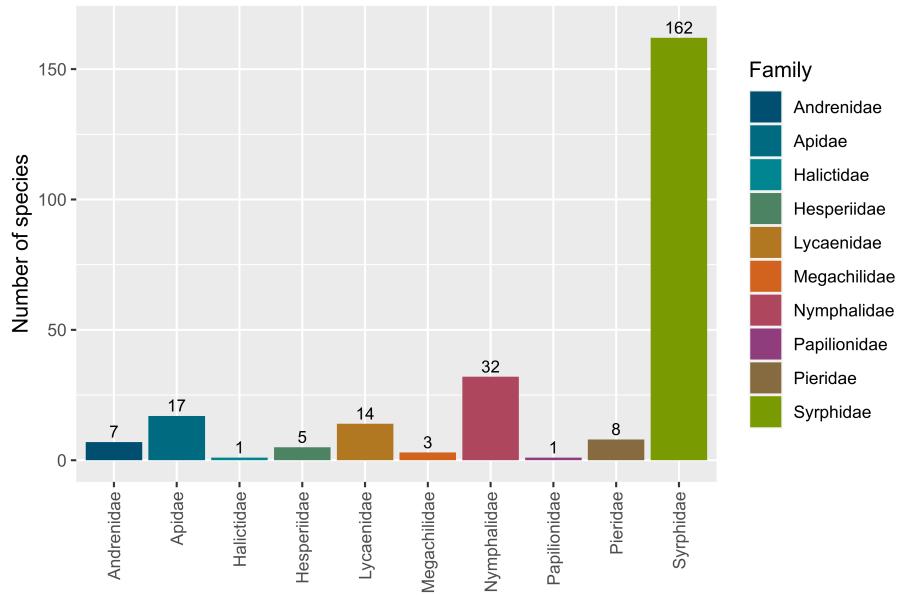
```

`summarise()` ungrouping output (override with `groups` argument)

```

ggplot(poll_spec_richn, aes(x = family, y = col_spec_richn, label = col_spec_richn)) +
  geom_bar(stat = "identity",
            aes(fill = family)) +
  geom_text(size = 3,
            position = position_nudge(y = 4)) +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1)) +
  scale_fill_nina(name = "Family") +
  xlab("") +
  ylab("Number of species")

```



```

pollinator_species <- dat %>%
  filter(trap_type == "Malaise") %>%
  filter(family == "Andrenidae" |
            family == "Apidae" |
            family == "Colletidae" |
            family == "Halictidae" |
            family == "Megachilidae" |
            family == "Melittidae" |
            family == "Syrphidae" |

```

```

family == "Hesperiidae" |
family == "Lycaenidae" |
family == "Nymphalidae" |
family == "Papilionidae" |
family == "Pieridae" |
family == "Riodinidae"
) %>%
distinct(order, family, genus, species) %>%
arrange(family, genus, species)

dat %>%
filter(trap_type == "Malaise") %>%
filter(family == "Apidae") %>%
distinct(genus, species) %>%
select(genus, species)

## # A tibble: 17 x 2
##   genus     species
##   <chr>    <chr>
## 1 Apis      mellifera
## 2 Bombus    soroeensis
## 3 Bombus    pascuorum
## 4 Bombus    lucorum
## 5 Bombus    jonellus
## 6 Bombus    pratorum
## 7 Bombus    monticola
## 8 Melipona  illota
## 9 Bombus    consobrinus
## 10 Bombus   cryptarum
## 11 Bombus   sporadicus
## 12 Bombus   cingulatus
## 13 Bombus   hortorum
## 14 Bombus   hypnorum
## 15 Bombus   norvegicus
## 16 Bombus   ashtoni
## 17 Bombus   terrestris
write_csv(pollinator_species,
          path = "out/pollinator_species.csv")

```

Phenology graphs

Some “timeseries” throughout the season.

```

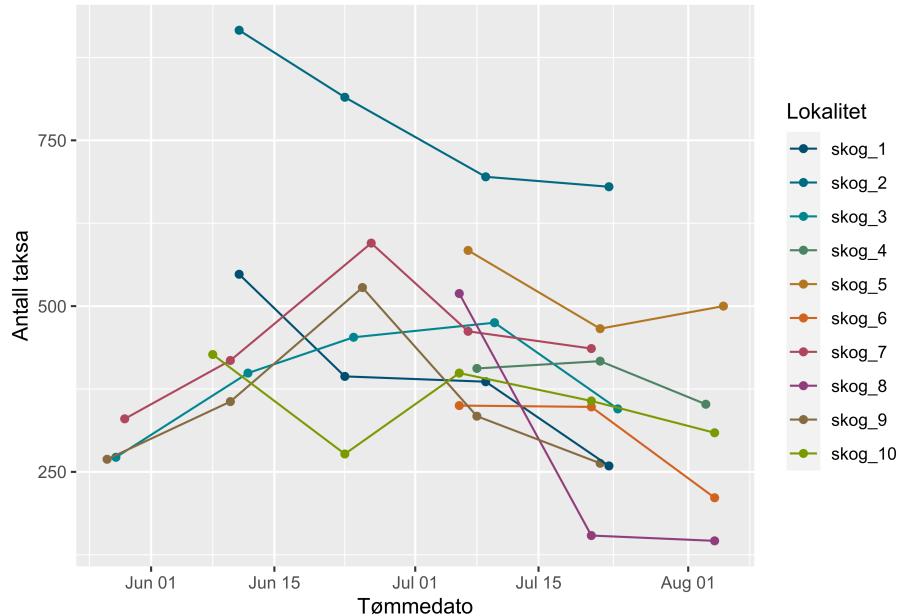
seasonAvgNo <- dat %>%
  filter(weeks_sampled == 2) %>%
  group_by(habitat_type, date_emptied, locality) %>%
  summarise(no_species = n_distinct(genus, species),
            sum_biomass = sum(Vekt_flaske_..insekter_vaat_.g., na.rm = T) / no_species)

## `summarise()` regrouping output by 'habitat_type', 'date_emptied' (override with `groups`)

# test <- seasonAvgNo <- dat %>%
#   filter(weeks_sampled == 2) %>%
#   group_by(habitat_type, date_emptied, locality) %>%
#   select(habitat_type, date_emptied, locality, Vekt_flaske_..insekter_vaat_.g.) %>%
#   distinct() %>%
#   summarise(sum(Vekt_flaske_..insekter_vaat_.g., na.rm = T))

seasonAvgNo %>%
  filter(habitat_type == "Skog") %>%
  ggplot() +
  geom_point(aes(x = date_emptied, y = no_species, color = locality)) +
  geom_line(aes(x = date_emptied, y = no_species, color = locality)) +
  ylab("Antall taksa") +
  xlab("Tømmedato") +
  scale_color_nina(name = "Lokalitet")

```



```

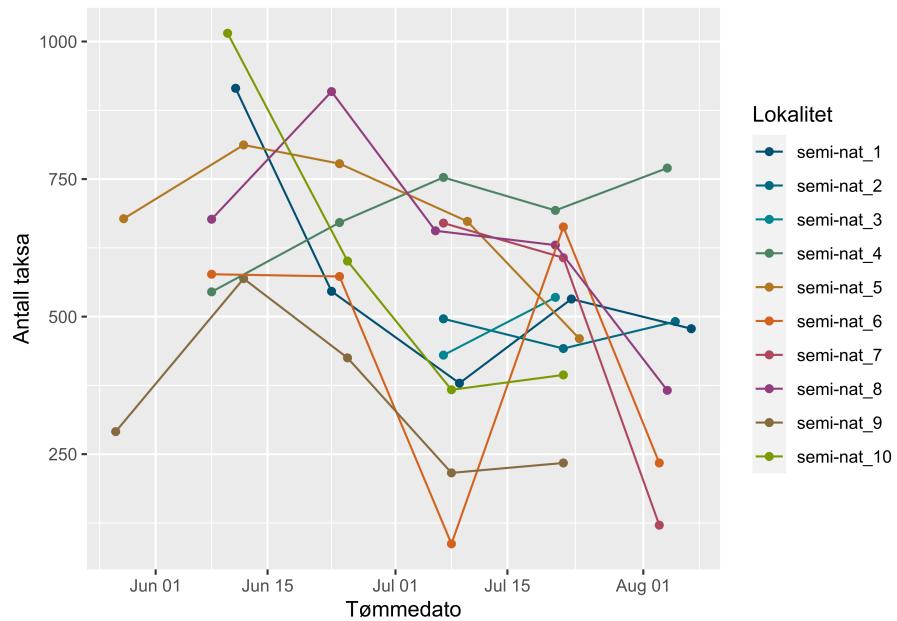
seasonAvgNo %>%
  filter(habitat_type == "SN_Gressmark") %>%

```

```

ggplot() +
  geom_point(aes(x = dateemptied, y = nospecies, color = locality)) +
  geom_line(aes(x = dateemptied, y = nospecies, color = locality)) +
  ylab("Antall taksa") +
  xlab("Tømmedato") +
  scale_color_nina(name = "Lokalitet")

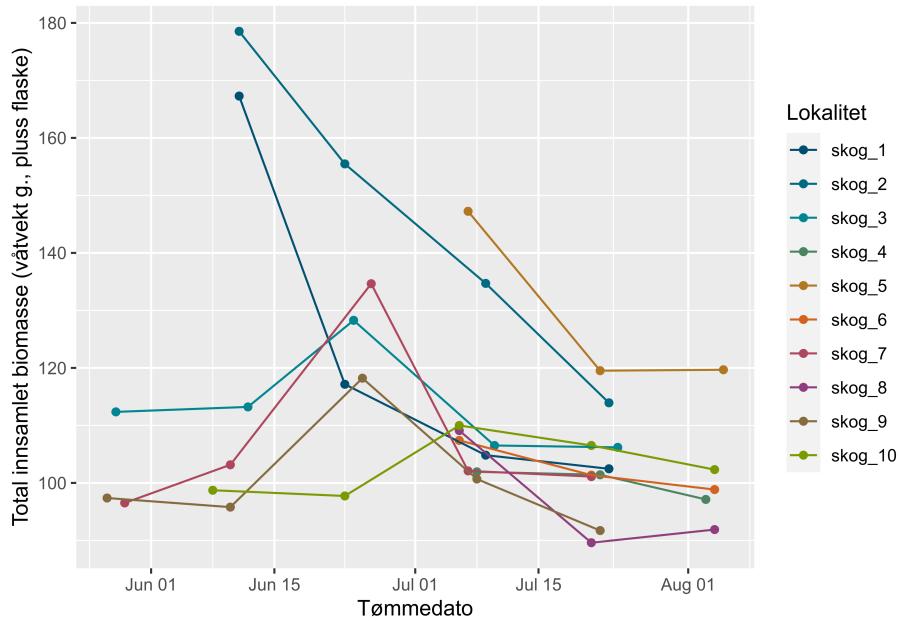
```



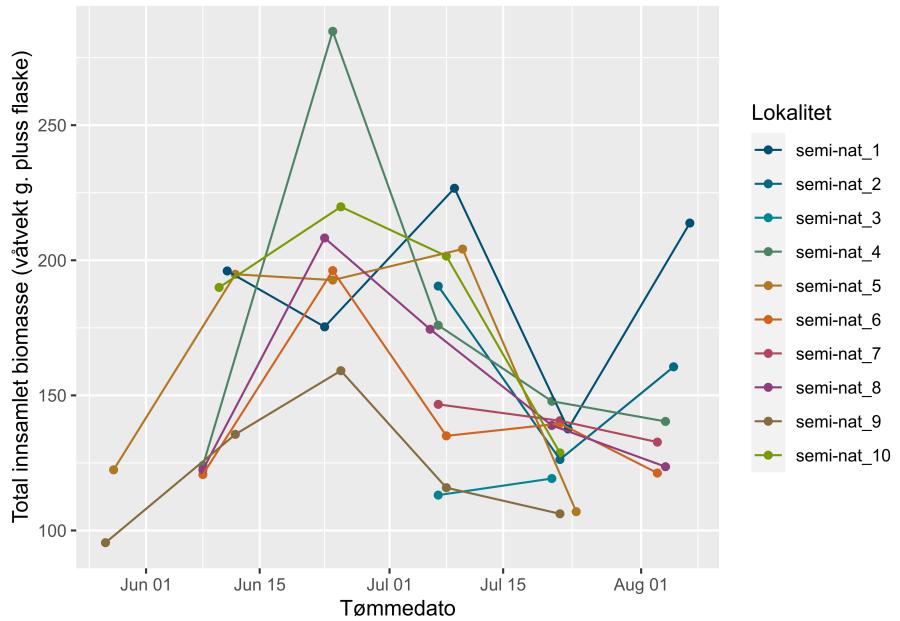
```

seasonAvgNo %>%
  filter(habitat_type == "Skog") %>%
  ggplot() +
  geom_point(aes(x = dateemptied, y = sumbiomass, color = locality)) +
  geom_line(aes(x = dateemptied, y = sumbiomass, color = locality)) +
  ylab("Total innsamlet biomasse (våtvekt g., pluss flaske)") +
  xlab("Tømmedato") +
  scale_color_nina(name = "Lokalitet")

```



```
seasonAvgNo %>%
  filter(habitat_type == "SN_Gressmark") %>%
  ggplot() +
  geom_point(aes(x = dateemptied, y = sum_biomass, color = locality)) +
  geom_line(aes(x = dateemptied, y = sum_biomass, color = locality)) +
  ylab("Total innsamlet biomasse (våtvekt g. pluss flaske)") +
  xlab("Tømmedato") +
  scale_color_nina(name = "Lokalitet")
```



Plot some weather data

```
logger_tbl <- tbl(con,
                   in_schema("loggers", "logger_data")) %>%
  select(-id)

logger_dep_tbl <- tbl(con,
                      in_schema("loggers", "logger_deployments")) %>%
  select(-id)

logger_data <- logger_tbl %>%
  left_join(logger_dep_tbl,
            by = c("logger_id" = "logger_id",
                  "logger_type" = "logger_type")) %>%
  mutate(day = as.Date(date))
```

Plot average values for each site during July

```
tt <- logger_data %>%
  group_by(location, day, data_type) %>%
  summarize(value = mean(value)) %>%
  filter(day >= '2020-07-01' &
         day < '2020-08-01'
```

```

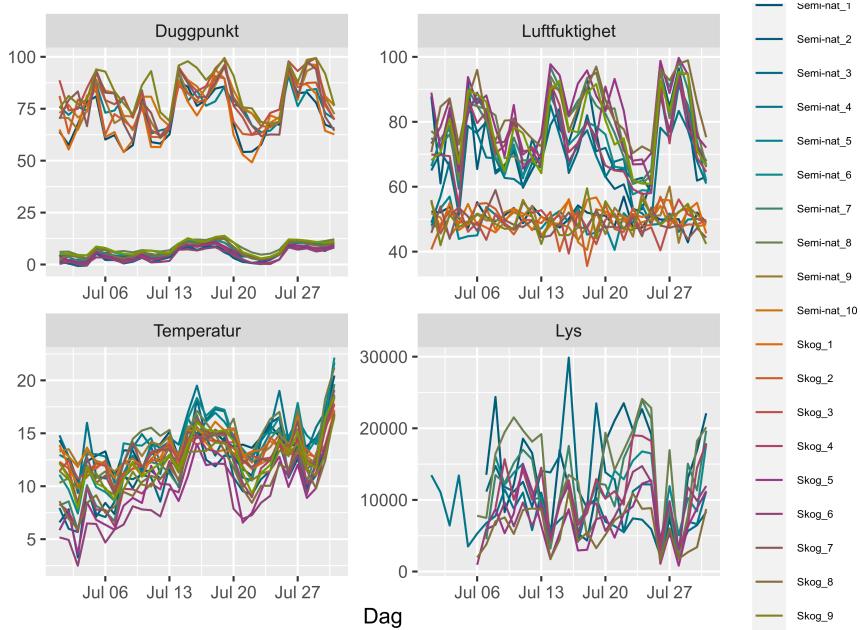
) %>%
collect() %>%
mutate(Lokalitet = factor(location, levels = c(paste0("Semi-nat_", 1:10), paste0("Skog_", 1:9)),
mutate(data_type = factor(data_type, levels = )))

## Warning: Missing values are always removed in SQL.
## Use `mean(x, na.rm = TRUE)` to silence this warning
## This warning is displayed only once per session.

## Why can't I set the levels in the pipe?
levels(tt$data_type) <- c("Duggpunkt", "Luftfuktighet", "Temperatur", "Lys")

tt %>%
ggplot(.) +
geom_line(aes(x = day, y = value, color = Lokalitet)) +
facet_wrap(~data_type,
           scales = "free") +
scale_color_nina(name = "Lokalitet") +
xlab("Dag") +
ylab("") +
theme(legend.text=element_text(size=rel(0.5)))

```



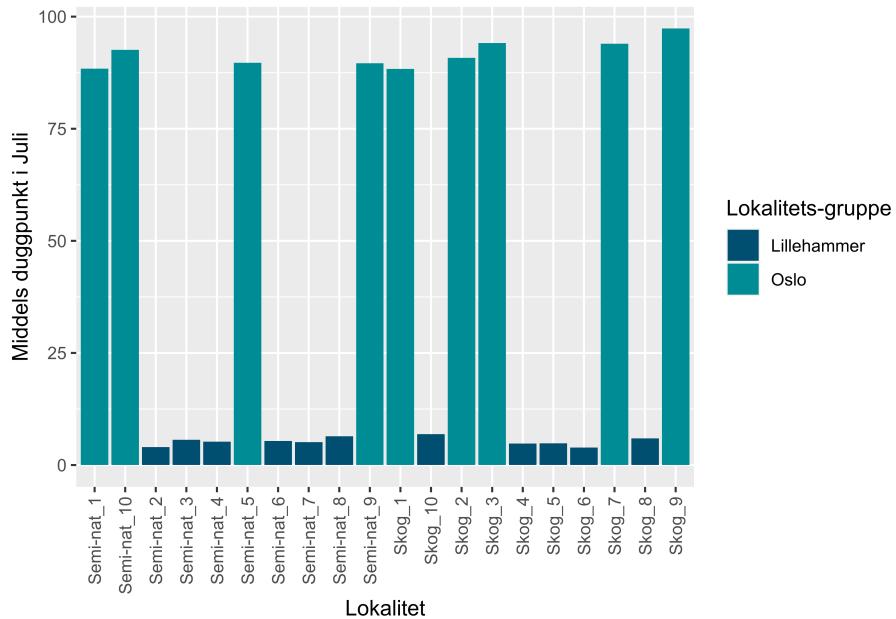
There seems to be two clusters of relative humidity values. They seem to be different in the two location groups (Oslo, Lillehammer). Where they placed differently? Or is it just altitude? Must plot this with altitude as well.

```

location_groups <- tibble(location = c(paste0("Semi-nat_", 1:10), paste0("Skog_", 1:10)),
                           location_group = c("Oslo", "Lillehammer", "Lillehammer", "Lilleham
                           "Oslo", "Lillehammer", "Lillehammer", "Lilleham
                           "Oslo", "Oslo", "Oslo", "Oslo", "Oslo", "Oslo", "Lilleh
                           "Lillehammer", "Lillehammer", "Oslo", "Lilleham
                           "Oslo", "Lillehammer"))

logger_data %>%
  filter(day >= '2020-10-01' &
        day < '2020-11-01') %>%
  group_by(location, data_type) %>%
  summarize(value = mean(value)) %>%
  filter(data_type == "dew") %>%
  collect() %>%
  mutate(location = factor(location, levels = c(paste0("Semi-nat_", 1:10), paste0("Skog_", 1:10)),
                           left_join(location_groups,
                                     by = c("location" = "location")) %>%
  ggplot(.) +
  geom_bar(aes(x = location, y = value, fill = location_group), stat = "identity") +
  scale_fill_nina(name = "Lokalitets-gruppe") +
  xlab("Lokalitet") +
  ylab("Middels duggpunkt i Juli") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))

```



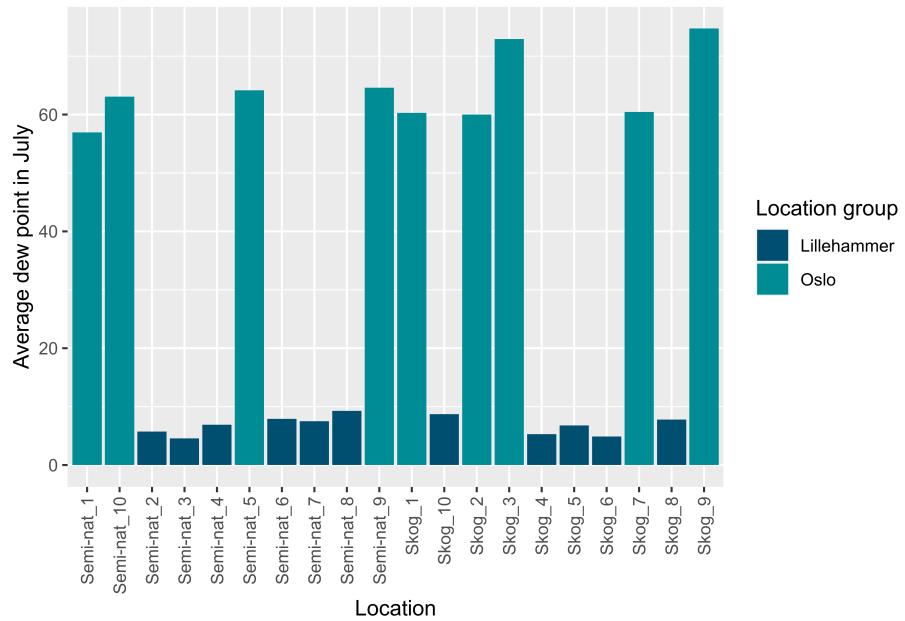
What happens if we only take the day time values?

```

logger_data %>%
  filter(day >= '2020-07-01' &
        day < '2020-08-01') %>%
  collect() %>%
  mutate(time = as_hms(date)) %>%
  filter(time >= as_hms("12:00:00") &
         time < as_hms("18:00:00")) %>%
  filter(data_type == "dew") %>%
  group_by(location) %>%
  summarize(value = mean(value)) %>%
  mutate(location = factor(location, levels = c(paste0("Semi-nat_", 1:10), paste0("Skog_", 1:10))) %>%
  left_join(location_groups,
            by = c("location" = "location")) %>%
  ggplot(.) +
  geom_bar(aes(x = location, y = value, fill = location_group), stat = "identity") +
  scale_fill_nina(name = "Location group") +
  xlab("Location") +
  ylab("Average dew point in July") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))

## `summarise()` ungrouping output (override with `.`groups` argument)

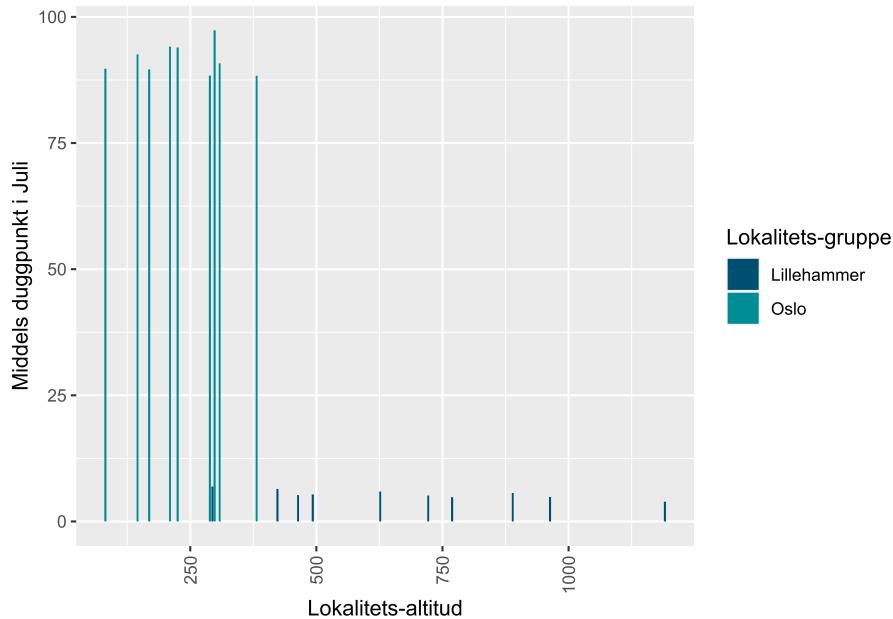
```



Check correspondance with altitude

```
location_alt_q <- "
SELECT location, avg(ST_Z(geom)) avg_altitude
FROM traps.trap_locations
WHERE elev_data IS TRUE
GROUP BY location
"
location_alt <- dbGetQuery(con, location_alt_q)

logger_data %>%
  filter(day >= '2020-10-01' &
        day < '2020-11-01') %>%
  group_by(location, data_type) %>%
  summarize(value = mean(value)) %>%
  filter(data_type == "dew") %>%
  collect() %>%
  mutate(location = factor(location, levels = c(paste0("Semi-nat_", 1:10), paste0("Skog_", 1:10)),
                           ordered = TRUE)) %>%
  left_join(location_groups,
            by = c("location" = "location")) %>%
  left_join(location_alt,
            by = c("location" = "location")) %>%
  filter(!is.na(avg_altitude)) %>%
  ggplot(.) +
  geom_bar(aes(x = avg_altitude, y = value, fill = location_group), stat = "identity") +
  scale_fill_nina(name = "Lokalitets-gruppe") +
  xlab("Lokalitets-altitud") +
  ylab("Middels duggpunkt i Juli") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```



The dew points seem to correspond quite well with the altitude,

```
logger_data %>%
  filter(day >= '2020-10-01' &
        day < '2020-11-01') %>%
  group_by(location, data_type) %>%
  summarize(value = mean(value)) %>%
  filter(data_type == "rh") %>%
  collect() %>%
  mutate(location = factor(location, levels = c(paste0("Semi-nat_", 1:10), paste0("Skog_", 11:20))),
         location_group = case_when(location %in% c("Semi-nat_1", "Semi-nat_2", "Semi-nat_3", "Semi-nat_4", "Semi-nat_5", "Semi-nat_6", "Semi-nat_7", "Semi-nat_8", "Semi-nat_9", "Semi-nat_10") ~ "Semi-nat",
                                      location %in% c("Skog_11", "Skog_12", "Skog_13", "Skog_14", "Skog_15", "Skog_16", "Skog_17", "Skog_18", "Skog_19", "Skog_20") ~ "Skog"))
  left_join(location_groups,
            by = c("location" = "location")) %>%
  left_join(location_alt,
            by = c("location" = "location")) %>%
  filter(!is.na(avg_altitude)) %>%
  ggplot(.) +
  geom_bar(aes(x = avg_altitude, y = value, fill = location_group), stat = "identity") +
  scale_fill_nina(name = "Lokalitets-gruppe") +
  xlab("Lokalitets-altitud") +
  ylab("Middels duggpunkt i Juli") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```

